**Standard Operating Procedure (SOP) emR analysis**

**Tab. 1: Specification of emR functions.**

|  |  |
| --- | --- |
| **Function** | **Description** |
| add\_median\_survival | * Fitting a survival fit for each subgroup defined by var and extracting median survival times * Additionally the total survival time for the whole sample is calculated * Is called within the function “outcome\_table\_survival” |
| barplot\_eumelareg | * Generates Barplot in predefined emR style * Default color is “jco” color palette |
| calc\_DCR | * Calculate disease control rate based on a variable containing information about best response |
| calc\_ORR | * Calculate overall response rate based on a variable containing information about best response |
| calc\_sankey\_flow | * Transform data frame into format, which can be used in the function “sankey\_plot\_eumelareg” |
| calc\_survival | * Calculates survival time based on an event start and end date |
| convert\_date | * Converts date into standard yyyy-mm-dd format |
| convert\_response | * Transform best response variable to uniform format (e.g. NED is transformed to CR) |
| cox\_output | * Calculates cox regression * Automatic distinction between univariate and multivariate based on variable input (if vars argument > 1 multivariate result is printed) * Also allows for backwards selection of a multivariate cox model |
| cox\_table | * Wrapper around “cox\_output” to print output in a table |
| cox\_table\_combined | * Wrapper around “cox\_table” to print univariate, multivariable and reduced multivariable cox regression |
| emr\_cols | * Color scheme with the emr colors * Deprecated as “jco” colors are now used |
| forestplot\_eumelareg | * Altered version of ggforest from the survminer package |
| get\_stats | * Get description statistics for survival analysis |
| outcome\_table\_survival | * generates an outcome table with data for best response, dcr, orr and median PFS, OS and ToT |
| pie\_chart\_eumelareg | * draws pie chart with emr specifications |
| sankey\_plot\_eumelareg | * generates Sankey plot out of input generated with “calc\_sankey\_flow” |
| survplot\_eumelareg | * draws Kaplan-meier survival curves * also adds median survival table and number of patients at risk * survival table and KM-plot can be printed in one figure or separately based on the argument “merge” |
| theme\_eumelareg\_barplot | * theme for barplots in eumelareg style |
| theme\_eumelareg\_base | * base theme for eumelareg plots |
| theme\_eumelareg\_pie\_chart | * theme for pie charts in eumelareg style |
| theme\_eumelareg\_surv\_plot | * theme for Kaplan-meier plots in eumelareg style |
| theme\_eumelareg\_surv\_table | * theme for risk table within Kaplan-meier plot |

**How to use the functions and modify output generated with these functions:**

**Tables:**

* generate output in R
* in the viewer panel click “show in new window”
* copy the content of the webpage into excel
* format rows and columns
  + set width of columns as wide as the longest entry in each column to avoid columns being merged
* copy content into power point and adjust size if needed

**Survival curves:**

* generate output with “survplot\_eumelareg”
* formatting:
  + adjust plot height with “plot.height” (relative value between 0 and 1)
  + set width of the risk table with “risk.table.width”
  + move plot with “plot.margin.left” to align plot with risk table
  + save plot:
    - png(paste(path, filename,"\_plot.png", sep = ""), units="in", width=9, height=6.2, res=600)
    - code to plot figure
    - dev.off()
      * units can be inches (“in”), millimeters (“mm”) or pixels (“px”)
      * res: resolution, set to at least 300
      * adjust size with “width” and “height”

**Forestplot:**

* generate output with forestplot\_eumelareg
* formatting:
  + set size of text with “fontsize”, default is 0.7
  + adjust columns with “cpositions = c(a, b, c)”
    - a, b, c: numbers between 0 and 1
    - a: position of row title
    - b: position of factor levels
    - c: position of HR (95% CI)
  + to adjust the row height parameters in the png function can be used
    - png(paste(path, filename,"\_plot.png", sep = ""), units="in", width=12, height=8, res=600)
    - code to plot figure
    - dev.off()